写在前面的话：

总结一下使用word2vec一年来的一些经验，因为自己在做的时候，很难在网上搜到word2vec的经验介绍，所以归纳出来，希望对读者有用。

这里不介绍word2vec的原理，因为原理介绍的资料网上很多：

作者论文 word2vec论坛

作者论文讲的比较简单，推荐一个比较全面的word2vec原理分析

最后，由于本人知识有限，错误之处，还望指正。

1 word2vec 是word embedding 最好的工具吗?

word2vec并非是效果最好的word embedding 工具。最容易看出的就是word2vec没有考虑语序，这里会有训练效果损失。

由于 word2vec 训练速度快 ，易用，google出品 等，使得word2vec使用的人多。

训练快是因为 word2vec只有输入层和输出层，砍去了神经网络中，隐藏层的耗时计算（所以word2vec并不算是一个深度学习算法）。另外，阅读word2vec的google的源码，会发现里面有一些提速的trick。如 sigmod函数，采用一次计算，以后查表，减去了大量的重复计算。如词典hash存储， 层次softmax等。

易用是因为word2vec 公布了word2vec的代码。在tensorflow,gensim,spark mllib包中都有集成，使用方便。

2 word2vec 训练结果的差异主要来自什么因素?

2.1 语料影响最大

语料的场景，比如微博的语料和新闻语料训练的结果差别很大。因为微博属于个人发帖，比较随意。而新闻比较官方正式，另外新闻句式相对复杂。经过训练对比：微博这种短文，训练的相似词更多是同级别的相关词。比如 深圳 相关的是 广州 。而用新闻语料，训练得到 深圳 相关的词 更多是与 深圳 有关联的词，比如 深圳大学。

实际发现在微博，违法色情的词训练的比较好，因为黑产用这种聚到来推广。而在评论，骂人的词训练的比较好，在新闻，则是常见的正规的词训练的比较好。


input word EXIT for exit:垃圾
辣鸡 0.775921
拉圾 0.632468
腊鸡 0.617436
狗屎 0.600743
废物 0.559560
渣子 0.553082
sb 0.551927
傻逼 0.541273
臭狗屎 0.538914
狗逼 0.538713
脑残 0.535560
渣滓 0.532699
玩意 0.517474
沙比 0.515293
货色 0.513036
垃 0.502785
烂 0.502723
tmd 0.497506
圾 0.497001
他妈的 0.496971
不入流 0.495143
烂货 0.494959
渣渣 0.489287
恶心 0.488262
狗币 0.487922
傻比 0.486765
玩意儿 0.483344
狗东西 0.482354
臭不要脸 0.481913
破烂货 0.480917

评论 垃圾 的相关词

新闻 垃圾 的相关词

为什么会出现这种情况呢?

因为 word2vec 的原理就是 一个词预测 前后词 或者 前后词 预测 当前词，使得概率最大化。

这就导致如下两个结果：

2.1.1 相似的句子，相同部位的词 会相似。

比如 句子1 w1 w2 w3 w4 X w5 w6 w7.

句子2 w1 w2 w3 w5 Y w5 w6 w7.

因为 X 的向量 受 w1 w2 w3 w4 w5 w6 w7 向量影响决定， Y也是受这几个词影响决定。

所以 X Y 是相似的。

2.1.2 挨着近的词，也是相似的。

比如 句子 w1 w2 w3 w4 X Y w5 w6 w7. 这样 X Y 都是受到 来自 w1 w2 w3 w4 w5 w6 w7 向量影响决定。

所以X Y是相似的。

所以，微博和新闻的句子的整体分布是不一样的。 这里影响 结论 2.1.1.

其次，新闻长文多，句式复杂，微博短文多，这里影响结论2.1.2.

2.2 算法参数的影响。

算法参数对总体效果影响不大。相对来说，比较重要的参数有以下：

2.2.1 将采样(subsampling)

降采样越低，对高频词越不利，对低频词有利。可以这么理解，本来高频词 词被迭代50次，低频词迭代10次，如果采样频率降低一半，高频词失去了25次迭代，而低频词只失去了5次。一般设置成le-5。个人觉得，降采样有类似tf-idf的功能，降低高频词对上下文影响的权重。

2.2. 2 语言模型

skip-gram 和cbow,之前有对比，切词效果偏重各不相同。 从效果来看，感觉cbow对词频低的词更有利。这是因为 cbow是基于周围词来预测某个词，虽然这个词词频低，但是他是基于 周围词训练的基础上，通过算法来得到这个词的向量。通过周围词的影响，周围词训练的充分，这个词就会收益。

2.2. 3 窗口大小

窗口大小影响 词 和前后多少个词的关系，和语料中语句长度有关，建议可以统计一下语料中，句子长度的分布，再来设置window大小。一般设置成8。

2.2. 4 min-count

最小词频训练阀值，这个根据训练语料大小设置，只有词频超过这个阀值的词才能被训练。

根据经验，如果切词效果不好，会切错一些词，比如 “在深圳”，毕竟切错的是少数情况，使得这种错词词频不高，可以通过设置相对大一点的 min-count 过滤掉切错的词。

2.2. 5 向量维度

如果词量大，训练得到的词向量还要做语义层面的叠加，比如 句子 的向量表示 用 词的向量叠加，为了有区分度，语义空间应该要设置大一些，所以维度要偏大。一般 情况下200维够用。

2.2. 6 其他参数

比如学习率 可以根据需要调。

3 word2vec 影响速度的因素有哪些?

3.1 语言模型：cbow 比skip-gram 更快

为什么 cbow更快，很重要的一个原因，cbow是基于周围词来预测这个单词本身 。而skip-gram是基于本身词去预测周围词。 那么，cbow只要 把窗口内的其他词相加一次作为输入来预测 一个单词。不管窗口多大，只需要一次运算。而skip-gram直接受窗口影响，窗口越大，需要预测的周围词越多。在训练中，通过调整窗口大小明显感觉到训练速度受到很大影响。

3.2 迭代次数

影响训练次数，语料不够的情况下，可以调大迭代次数。spark 版本有bug，迭代次数超过1，训练得到的词向量维度值超大。

3.3 线程数

单机版（google word2vec)可以通过设置多线程跑,集群版（spark mllib）可以设置多个 partitions.但是从经验来看，在集群上设置partitions 过多，会影响训练的效果。

3.4 其他参数

采样频率 影响词的训练频率 min-count 最小词频 影响 训练词的数量 Window大小 影响 skip-gram 的预测次数。 向量维度 维度决定了训练过程中计算的维度

4 怎样评估word2vec训练的好坏?

4.1 词聚类

可以采用 kmeans 聚类，看聚类簇的分布

4.2 词cos 相关性

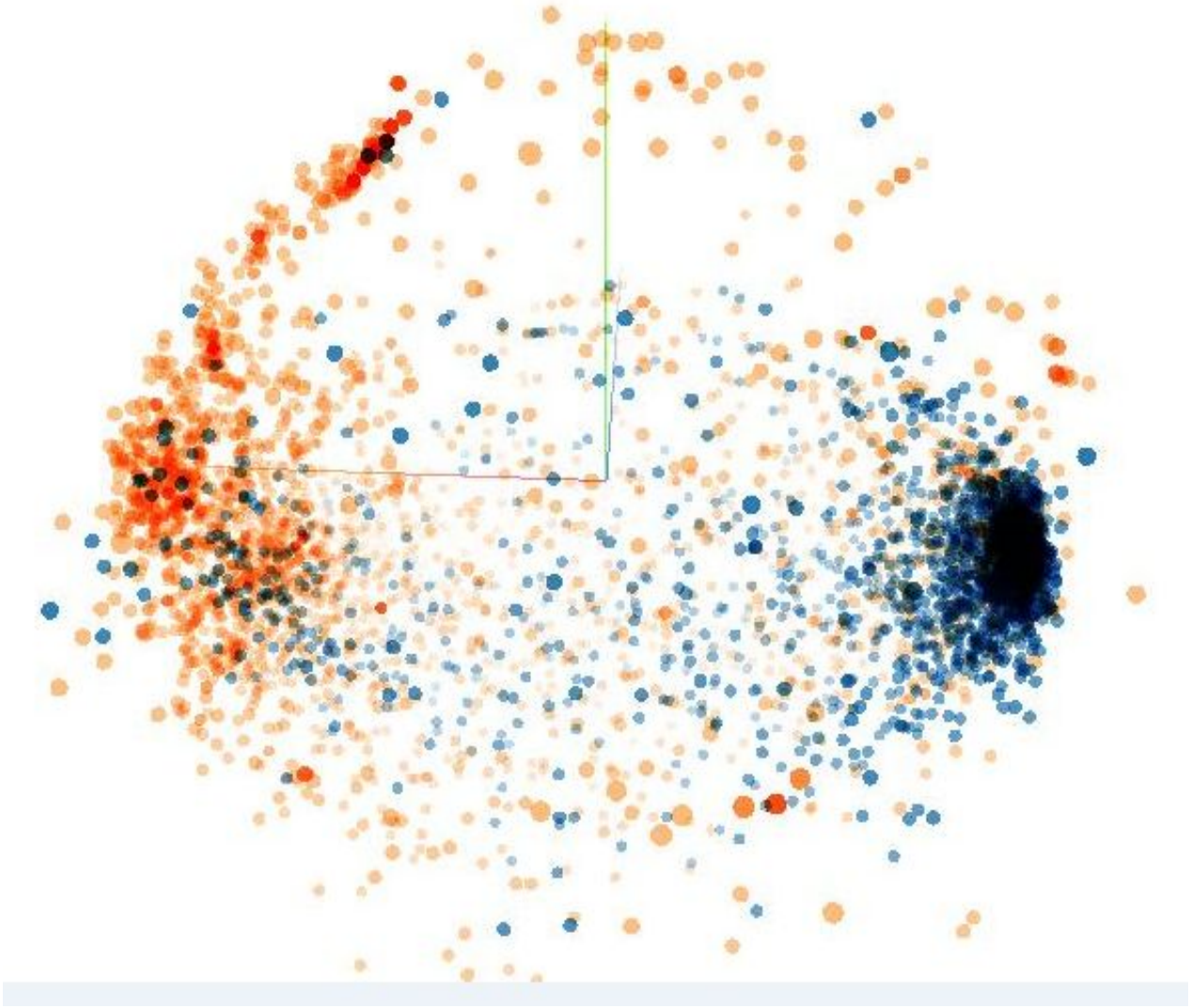
查找cos相近的词

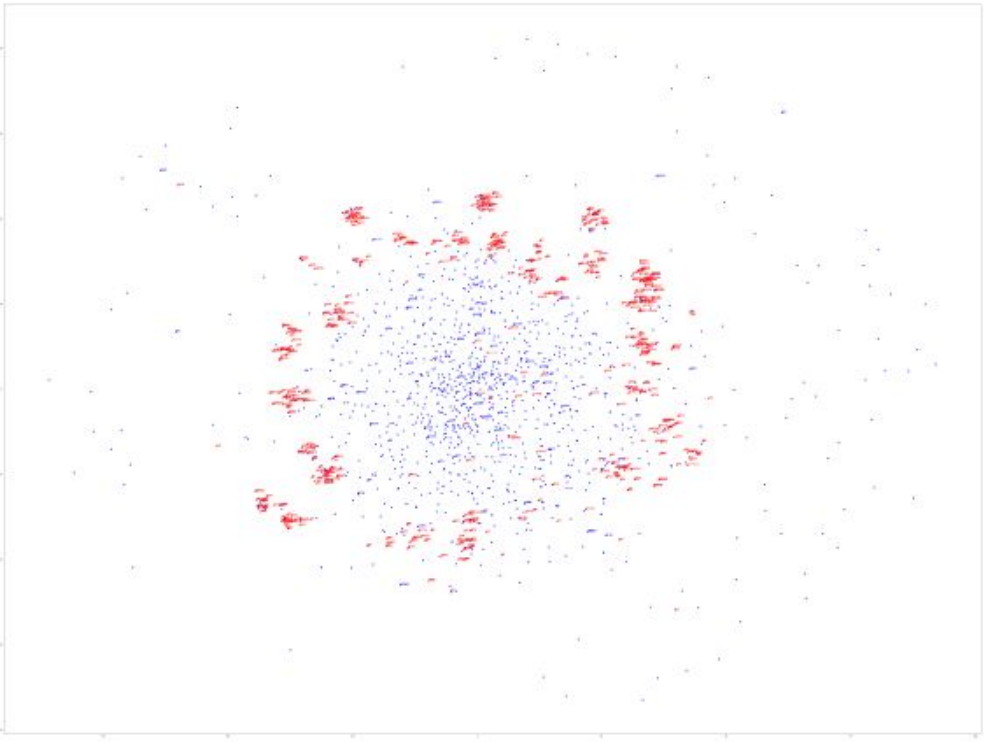
4.3 Analogy对比

a:b 与 c:d的cos距离 (man-king woman-queen)

4.4 使用tnse，pca等降维可视化展示

词的分布，推荐用google的tensorboard，可以多视角查看，如果不想搭建服务，直接访问这里。另外可以用python的matplotlib。





4.5 Categorization 分类 看词在每个分类中的概率

词动物食物汽车电子橘子0.110.680.120.11鸟0.660.110.130.11雅阁0.140.230.670.11苹果0.110.650.110.65

前三条来自官网的评测方法

网上也有相关的word embedding 的评估方法，可以参考这里

资源

- word2vec :https://github.com/zhyq/word2vec-google
- spark mllib word2vec训练，转换成google word vector形式:https://github.com/zhyq/word2vec-spark

word2vec效果展示

弟弟	
姐姐	0.822271
哥哥	0.739340
儿子	0.721118
外甥	0.719428
弟妹	0.717014
妹妹	0.700240
姐弟	0.690913
爸爸	0.678089
女儿	0.674342
我的小	0.673382
姐弟俩	0.663059
小叔	0.660328
表弟	0.657854
两姐妹	0.657631
亲妹妹	0.652114
姑姑	0.651611
亲弟弟	0.649457
父亲	0.636804
亲姐姐	0.635719
姐妹俩	0.635299
弟媳妇	0.634309
侄子	0.630858
妈妈	0.630187
二姐	0.622875
嫂子	0.620430
二哥	0.620051
弟	0.618766
堂哥	0.617703
二妹	0.615990
姑妈	0.614280
表哥	0.613995
兄妹俩	0.613074
堂弟	0.611449
伐的妹妹	0.610482
两兄弟	0.601802
弟弟妹妹	0.600538
小儿子	0.600149
舅舅	0.595470
表姐	0.595226
母亲	0.592240
婶婶	0.592102

菠萝	
菠萝蜜	0.758285
凤梨	0.687275
芒果	0.665832
水果	0.632165
杨桃	0.631353
草莓	0.622794
橙子	0.617050
吃菠萝	0.615634
火龙果	0.614227
百香果	0.609204
椰子	0.598953
金桔	0.592254
番石榴	0.588913
香蕉	0.587047
圣女果	0.583899
猕猴桃	0.582926
柠檬	0.582876
莲雾	0.582790
小番茄	0.582102
哈蜜瓜	0.580340
凤梨汁	0.566840
番茄	0.566409
西瓜	0.566325
哈密瓜	0.561071
木瓜	0.558442
果肉	0.555012
菠萝汁	0.552229
香瓜	0.546142
榴莲	0.542659
桃子	0.542323
山竹	0.541995
葡萄	0.530717
青芒果	0.530077
香甜可口	0.524735
柚子	0.523615
荔枝	0.521768
橘子	0.519520
番荔枝	0.519239
台湾凤梨	0.518178
皇帝柑	0.517108
木菠萝	0.516755

跑男	
奔跑吧兄弟	0.802604
跑男团	0.791064
郑恺	0.714080
第四季	0.700071
极限挑战	0.697987
撕名牌	0.693247
挑战者联盟	0.664244
第三季	0.664000
偶像来了	0.649116
快乐大本营	0.648239
兄弟团	0.642272
陈赫	0.641199
鹿晗	0.632813
录节目	0.630010
王祖蓝	0.626934
极速前进	0.622950
综艺节目	0.612615
第一季	0.610054
节目效果	0.609364
大牌对王牌	0.608112
真人秀	0.603785
第二季	0.601148
季嘉宾	0.598801
大黑牛	0.595703
小猎豹	0.594894
丛林的法则	0.592832
节目组	0.590335
邓超	0.586581
女猎人	0.585754
录制节目	0.585597
金钟国	0.585260
伐木累	0.582880
傻狍子	0.578329
星星的密室	0.578238
西游奇遇记	0.576151
奔跑吧	0.575657
整季	0.573240
节目	0.571386
鹿	0.570864
韩国跑男	0.570825
男生女生向前冲	0.569674

板砖砸过来(^_^)

转载自：https://zhuanlan.zhihu.com/p/29364112